Production of morphological dictionaries of multi-word units using a multipurpose tool

Ranka Stanković and Ivan Obradović
University of Belgrade — Faculty of Mining and Geology, Djušina 7, 11000 Belgrade, Serbia
Email: {ranka,ivano}@rgf.bg.ac.rs

Cvetana Krstev
University of Belgrade — Faculty of Philology, Studentski trg 3, 11000 Belgrade, Serbia
Email: cvetana@matf.bg.ac.rs

Duško Vitas
University of Belgrade — Faculty of Mathematics, Studentski trg 16, 11000 Belgrade, Serbia
Email: vitas@matf.bg.ac.rs

***Abstract*—In this paper we outline the use of the multipurpose software tool LeXimir in our approach to automated production of lemmas for e-dictionaries of multi-word units. Development of morphological dictionaries of MWUs is a tedious task, especially in the case of Serbian and other languages featuring complex morphological structures. After realizing that the development of such a dictionary manually is an extremely slow process, we endeavored towards a procedure aimed at automated production of MWU dictionary lemmas, which is also outlined in this paper. The procedure was subsequently implemented as a new functionality of LeXimir, and makes use of our comprehensive e-dictionaries of Serbian simple words. We present an evaluation of the performance of this functionality, and hence of our procedure, obtained from experiments on two types of data. Finally, we discuss some further possible applications of our procedure and LeXimir in language processing tasks.**

I. Introduction

MORPHOLOGICAL electronic dictionaries of Serbian for natural language processing (NLP) are being developed for many years now. Their development follows the methodology and format (known as DELAS/DELAF) presented for French in [1]. E-dictionaries in the same format have been produced for many other languages. This format can be briefly described in the following way: in a dictionary of lemmas (DELAS) every lemma is described in full detail so that a dictionary of forms containing all necessary grammatical information (DELAF) can be generated from it. The dictionary of forms is used in NLP tasks. Two corpus processing systems that support work with this dictionary format were developed, Unitex [2] and Nooj [3], both of which use finite-state technology as initially introduced in [4]. Serbian e-dictionaries of simple forms have reached a considerable size: they have a total of more than 126,000 lemmas [5] generating more than 4.3 million forms. Unitex official web site contains a comprehensive list of references related to the production and usage of e-dictionaries for various languages while Unitex distribution contains large samples of e-dictionaries, including one for Serbian which covers a sample text, the Serbian translation of Voltaire's *Candide*.

Some multi-word compounds composed of two or more contiguous graphical words that show some degree of non-compositionality and have constant references can be described using a similar approach. The NLP community offered various approaches to lexical treatment of multi-word units (MWUs) that were analyzed in detail by Savary [6]. Productive classes of MWUs, like numerals and various named entities that rely on them (e.g. measurement phrases) can best be described by dictionaries in the form of finite-state transducers (FST), and a number of them were produced for Serbian as well [7]. Other contiguous MWUs that are idiosyncratic in nature, namely nouns and adjectives, have to be lexically described in a similar way as simple words. That means that a dictionary of MWU lemmas (DELAC) that is provided with information enabling the production of all inflected forms (DELACF) has to be developed. In practice this simple procedure is not easy to perform because MWU lemmas have to be collected, generated, and inflected.

II. Inflection of MWUs

In order to produce a list of MWU forms in a systematic way, it is necessary to decide what the lemma of all these forms is, what are its additional features, how do its simple word constituents inflect, and what is the inflectional behavior of a MWU as a whole. One can imagine that for some languages this complex procedure can be skipped and a list of MWU forms can be produced from scratch. Serbian is, however, like all Slavic languages a highly inflectional language and such a shortcut procedure cannot be applied. We will illustrate this with one example. The nominal MWU *petokraka zvezda* 'five-pointed star' consists of an adjective followed by a noun, which in Serbian is the natural order of an adjective and a noun in a MWU. However, this MWU, together with a few more allows a reverse order as well — *zvezda petokraka*. It is more often used in the singular, but it can be used in the plural as well. In Serbian, adjectives and nouns inflect in number and case, while adjective forms also depend on the gender, definiteness, comparison, and in some cases animacy. Adjectives and nouns do not inflect freely in a MWU — the values of categories for number, case and gender have to agree. The animacy is important only for the masculine gender nouns in the accusative singular, and since the gender of *zvezda* 'star' is feminine, the animacy is of no relevance for this MWU. Finally, as the adjective *petokrak* 'five-pointed' has no comparative and superlative forms, and

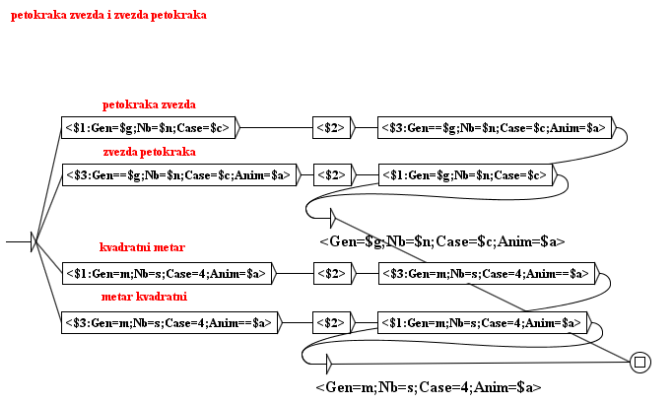


Fig. 1. A simplified transducer NC_AXNr for compounds of the type *petokraka zvezda* and *kvadratni metar*

definite and indefinite written adjective forms for feminine gender coincide, definiteness is of no relevance either.

This example illustrates the complexity of capturing all information about one MWU in its DELAC lemma. The most demanding part is to formulate the agreement conditions in a consistent way. A special form of inflectional transducers developed by Savary [8] and implemented in the Multiflex system answers most of these questions. The inflectional graph in Fig. 1 illustrates this. A MWU serving as lemma is tokenized and its tokens become values of variables, in our case $1=*petokraka*, $2=<space>, $3=*zvezda*. If a pattern of the form <$$i$ > appears in the inflectional graph it means that the corresponding token is recopied in all MWU inflectional forms as it is — in our example a second token, a space, is reproduced in all inflectional forms.

A token pattern can be followed by one or more equations of the type *Grammatical_feature=value*. In that case a specific form of a token is needed. In our example a token <$3:Gen=m;Nb=s;Case=4> from the lower part of the graph means that the masculine gender, singular and accusative form of the third token — the noun *zvezda* — is needed. However, the gender of the noun *zvezda* is feminine, so this form cannot be produced and the lower paths in the graph will be ignored. They will not be ignored for some other MWUs, like *kvadratni metar* 'square metar', since the gender of *metar* is masculine.

Additionally, grammatical-feature equations can contain not only concrete values but also unification variables. A unification variable instantiates to all values of the corresponding grammatical feature. For Serbian, a pattern <$3:Case=$c> means that forms for all cases — 7 different values — will be generated for the third token. The occurrence of the same unification variables in the same path means that their values have to agree. If a pattern <$1:Case=$c> appears in the same path as <$3:Case=$c> it means that when the genitive form of the first token is generated then the genitive form of the third token has to be generated as well, and that will also be the value of the 'Case' feature of the generated MWU form — the output of the transducer.

Finally, a unification variable does not need to instantiate to all values of some grammatical feature. Instead, it can inherit its value from a token itself. In the pattern <$3:Gen==$g> the variable $g inherits its value from the third token *zvezda* and instantiates only to the value *f* — the feminine gender. The variable $g from the pattern <$1:Gen=$g> occurring in the same path will thus have to agree with it and take the same value.

The two possible orders of the adjective and the noun in the MWU are achieved with two separate paths in the graph, one for the order given by a lemma itself, and the other for the reverse order. The orthographic variants of MWUs, e.g. the optional use of a hyphen, as well as omission of some of its constituents can be easily described using Multiflex graphs [9]. The Multiflex system is incorporated into Unitex, but it was also successfully used for Polish proper names in another environment [10].

By analogy with entries in a dictionary of simple word lemmas, an entry in a DELAC dictionary consists of a MWU lemma to which a name of an inflectional transducer (similar to the one represented in Fig. 1) is assigned. Similarity ends here, because simple word constituents of a MWU lemma also have to be described in a way that enables the production of all needed forms. This leads finally to the following lemma form:

```
petokraka(petokrak.A6:aefs1g)
zvezda(zvezda.N600:fs1q),NC_AXNr
```

This DELAC entry enables the production of 32 MWU forms for DELACF, one of which, representing the genitive singular with reverse order of constituents is:

```
zvezde petokrake,petokraka zvezda.N:fs2q
```

Production of a lemma in the format presented is far too demanding to be done manually because for each MWU one has to provide the following information:

1) What is the lemma? *petokraka zvezda*.
2) How does this MWU inflect and which inflectional transducer should be used for it? N_AXNr.
3) Which MWU constituents inflect? *petokraka* and *zvezda*.
4) What are DELAS entries of these MWU constituents that enable the generation of all needed forms? `petokrak.A6` and `zvezda.N600`.
5) What are the values of grammatical features of constituent forms used in the MWU lemma? `aefs1g` and `fs1q`.

The manual production of a lemma is, however, not necessary because possible answers to the above questions that concern MWU constituents can be found in dictionaries of simple words.

III. LeXimir as a Dictionary Management System

Bearing in mind the aforementioned complexity of production of MWU lemmas we have endeavored towards a procedure for automatic production of DELAC entries. The software tool which enabled the implementation of this procedure was

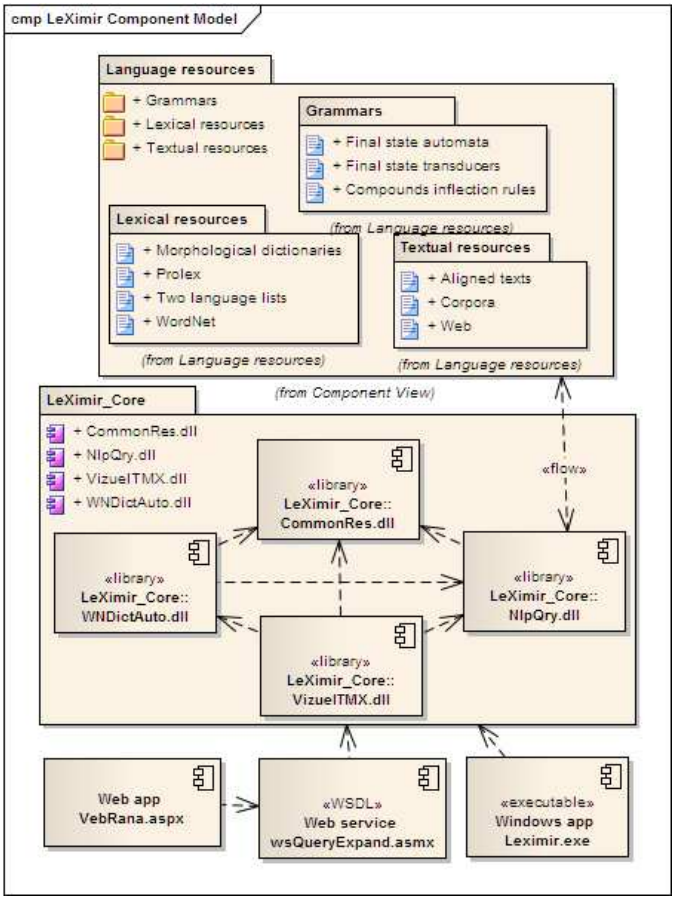


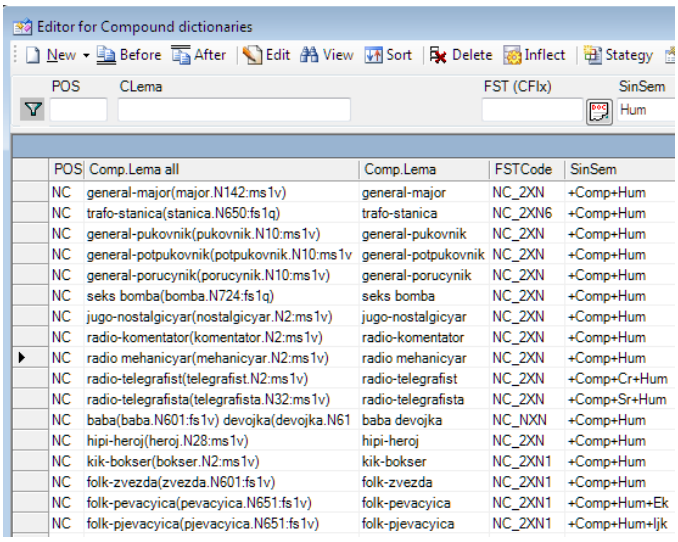
Fig. 2. Components of the software tool LeXimir

LeXimir, [1] a multipurpose tool developed by the University of Belgrade Language Technology Group [11] to support computational linguists in developing, maintaining and exploiting e-dictionaries. LeXimir is written in C#, and operates on the .NET platform. It can run on any personal computer under Windows and supports simultaneous manipulation of various language resources: e-dictionaries, wordnets, and aligned texts.

Implementation of LeXimir followed a modular approach. Namely, there exists a common core of the system, which is coupled with several modules performing different tasks. The central part of the system is *LeXimir_Core* composed of several .Net libraries: *CommonRes.dll*, *NlpQuery.dll*, *VisualTMX.dll* and *WNDictAuto.dll* (Fig. 2). For communication with lexical resources LeXimir makes use of the *NlpQuery.dll* module.

Modular organization of components provides two obvious benefits. In the first place, it enables the use of various resources in any part of the system, wherever they are needed. Thus, for example, morphological dictionaries can be used for adding additional morphological information to wordnet synsets, whereas both morphological dictionaries and the wordnet can be used in production of concordances for aligned

[1]LeXimir is available under CC NC BY licence. For more information see http://korpus.matf.bg.ac.rs/soft/LeXimir.html



POS	Comp.Lema all	Comp.Lema	FSTCode	SinSem
NC	general-major(major.N142:ms1v)	general-major	NC_2XN	+Comp+Hum
NC	trafo-stanica(stanica.N650:fs1q)	trafo-stanica	NC_2XN6	+Comp+Hum
NC	general-pukovnik(pukovnik.N10:ms1v)	general-pukovnik	NC_2XN	+Comp+Hum
NC	general-potpukovnik(potpukovnik.N10:ms1v	general-potpukovnik	NC_2XN	+Comp+Hum
NC	general-porucynik(porucynik.N10:ms1v)	general-porucynik	NC_2XN	+Comp+Hum
NC	seks bomba(bomba.N724:fs1q)	seks bomba	NC_2XN	+Comp+Hum
NC	jugo-nostalgicyar(nostalgicyar.N2:ms1v)	jugo-nostalgicyar	NC_2XN	+Comp+Hum
NC	radio-komentator(komentator.N2:ms1v)	radio-komentator	NC_2XN	+Comp+Hum
NC	radio mehanicyar(mehanicyar.N2:ms1v)	radio mehanicyar	NC_2XN	+Comp+Hum
NC	radio-telegrafist(telegrafist.N2:ms1v)	radio-telegrafist	NC_2XN	+Comp+Cr+Hum
NC	radio-telegrafista(telegrafista.N32:ms1v)	radio-telegrafista	NC_2XN	+Comp+Sr+Hum
NC	baba(baba.N601:fs1v) devojka(devojka.N61	baba devojka	NC_NXN	+Comp+Hum
NC	hipi-heroj(heroj.N28:ms1v)	hipi-heroj	NC_2XN	+Comp+Hum
NC	kik-bokser(bokser.N2:ms1v)	kik-bokser	NC_2XN1	+Comp+Hum
NC	folk-zvezda(zvezda.N601:fs1v)	folk-zvezda	NC_2XN1	+Comp+Hum
NC	folk-pevacyica(pevacyica.N651:fs1v)	folk-pevacyica	NC_2XN1	+Comp+Hum+Ek
NC	folk-pjevacyica(pjevacyica.N651:fs1v)	folk-pjevacyica	NC_2XN1	+Comp+Hum+Ijk

Fig. 3. LeXimir's editor for MWU dictionaries

texts. On the other hand, it enables the use of *LeXimir_Core* in different scenarios: as a stand alone Windows application LeXimir.exe or as a web application *VeBrana.aspx*[2], also known as VeBrana (previously WS4QE), which is supported by the *wsQueryExpand.asm* web service. The web service accepts and generates data sets in XML form, which are further converted into data structures that can be used for different purposes (string, array, table, etc.). As examples of web service functions we will mention a few characteristic ones: *getObliciLeme(lema)*, which generates inflected forms for a given lemma, *getSinonimiWN_WithFlex(lema)*, which returns all synonyms from a given wordnet synset in all inflected forms, and *getSinonimiWN_NoFlex(lema)* which returns synonyms without inflected forms.

As our e-dictionaries are Unitex-based, and Unitex is an open source software distributed under the LGPL license, we incorporated its modules in LeXimir for the majority of tasks that involve manipulation of e-dictionaries. For the production of MWU DELAC lemmas we used the appropriate Unitex modules for dictionary look-up.

LeXimir provides for concurrent manipulation of several dictionaries of lemmas, both of simple words and MWUs (DELAC), distributed in any number of files. However, the possibility of manipulating dictionaries of word forms is not envisaged, as such files are produced automatically either from DELAS or DELAC by means of appropriate FSTs. Organizing dictionaries in sets of different files is practically motivated. Namely, smaller size files are much easier to manipulate.

LeXimir's editor for MWUs is illustrated in Fig. 3. Besides the usual functions — add, insert, copy, change — a user can check the correctness of every lemma with the function 'Inflect' that lists all inflected forms of a selected lemma. Another useful function is the extraction of subsets of lemmas based on different criteria: lemmas' beginning, their part of

[2]http://hlt.rgf.bg.ac.rs/VebRana

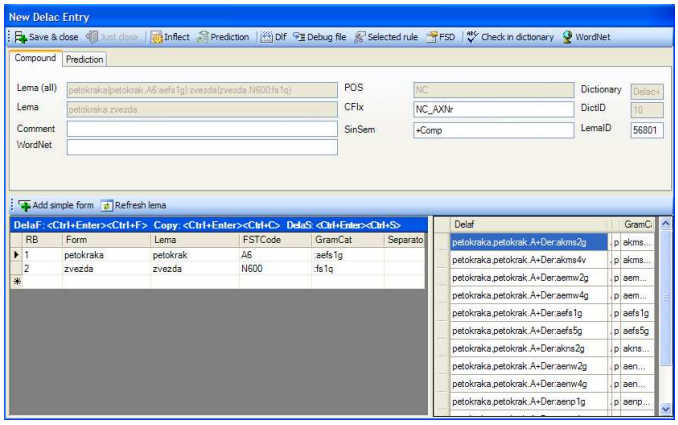


Fig. 4. The DELAC entry management form of Leximir

speech (PoS), inflectional class code, syntactic and/or semantic markers or a Boolean combinations of these criteria.

Figure 4 shows the table for manual production of a DELAC entry having two constituents: *petokraka* and *zvezda*. A user can insert constituents of a MWU in the column 'Form' of the table. In the next step columns 'Lemma', 'FST' (PoS and inflectional codes of constituents), and 'GramCat' (grammatical codes of constituents) have to be filled. The system does this automatically by offering all possible solutions retrieved from DELAS dictionaries of simple words. In the third step, the selection of the correct lemma, FST code and grammatical categories is supported by the possible combinations offered in auxiliary tables (in the right bottom corner of Fig. 4). In the final step, the user has to fill manually the code of the inflectional transducer for the newly produced MWU lemma, and attach to it the appropriate semantic and other markers. A user can then check the correctness of the new MWU lemma by using the 'Inflect' function that invokes Multiflex to perform the inflection.

The outlined procedure does help in answering the two last questions posed at the end of section II. However, answers to questions 2 and 3 have to be provided by the user. Thus, by following this approach not more than 2800 DELAC entries were produced during three years, which we found very ineffective.

IV. A Rule Based Procedure for Inflection of MWUs

A. Detection of inflectional properties of MWU lemmas

We have further improved the procedure for production of MWU lemmas when we realized that the answers obtained automatically in support of manual production of MWU lemmas can also help in detection of the syntactic composition of a MWU and therefore indicate the appropriate inflectional transducer. Namely, the MWUs in Serbian have predictable basic structures. For instance, nominal MWUs with two constituents (beside a separator) fall into five basic structures:

- Adjective/noun (both inflect and agree in gender, number and case)
- Noun/noun (both inflect and agree in number and case)
- Noun/noun in the genitive (only the first noun inflects)
- Word/noun (only the second noun inflects; the first word is usually not a Serbian simple word)
- Noun/adjective (both inflect and agree in gender, number and case)

However, there are 25 different inflectional graphs for the nominal MWUs with two constituents because there are subtleties that have to be taken into consideration besides these basic structures, e.g. can a MWUs have plural forms, can a separator be omitted or replaced by another separator, etc. The basic structure, however, determines the general form of a lemma and information that has to be supplied.

Thus, automatic production of the lemma for *petokraka zvezda* could proceed like this: a look-up in the dictionary of simple word forms determines that *zvezda* can only represent two realizations of the noun lemma *zvezda*, namely in the nominative singular or in the genitive plural. Similarly, it is determined that *petokraka* can be one of 12 different representations of the adjective *petokrak*; however, only one of them agrees with the noun *zvezda*, and that is the singular, feminine gender, nominative case form. Consequently, it can be deduced that only the basic structure adjective/noun applies here.

Of course, not all MWUs are so easy to process. For instance, for the MWU *vojna tajna* 'military secret' a dictionary look-up offers the following possibilities:

vojna	*vojni*	'military'	A	nom., sing., f.
vojna	*vojna*	'war'	N	nom., sing., f.
tajna	*tajna*	'secret'	N	nom., sing., f.
tajna	*tajni*	'secret'	A	nom., sing., f.

Thus there are three possible MWU structures: adjective/-noun, noun/noun and noun/adjective, whereas only the first one is correct.

Based on an analysis illustrated by the previous example, we have developed a new functionality within LeXimir that offers one or more DELAC entries for every MWU presented in its lemma form. As indicated by the example, it relies on information in e-dictionaries of simple words, but also uses a set of manually produced rules to deduce the basic structure of a given MWU, as well as its additional features. For the example *vojna tajna* this functionality would offer three lemmas; the first one would be selected, the other two discarded:

vojna(vojni.A2:aefs1g)	tajna(tajna.N6:fs1q)	AXN
vojna(vojna.N6:fs1q)	tajna(tajna.N6:fs1q)	NXN
vojna(vojna.N6:fs1q)	tajna(tajni.A5:aefs1g)	NXA

In order to design our automated procedure we grouped all inflectional transducers into equivalence classes or super-classes: a super-class consists of all MWUs having the same basic structure. It also means that their forms of MWU lemmas are the same because they need the same information for the production of inflectional forms. This is also reflected in the convention we used for naming the inflectional transducers: A stands for an adjective constituent, N stands for a noun

TABLE I
SUPER-CLASS AXN

Class	Example	Specifics
AXN	*vojna tajna*	
AXN3	*Ajfelova kula*	does not inflect in number
AXNF	*duhovni vodja*	second constituent changes gender in plural forms
AXNr	*petokraka zvezda*	allows reverse order

constituent, X stands for a constituent that does not inflect (including a separator), with some additional digits and letters added to differentiate transducers. This is illustrated in Table I by four classes (names of inflectional transducers) all belonging to the same AXN super-class and used for the inflection of MWUs consisting of an adjective followed by a noun, where both constituents inflect and must agree in basic grammatical categories.

In order to formulate a strategy for the production of MWU lemmas we analyzed the data available in the existing DELAC dictionary looking for useful information. On the one hand, we identified the additional information assigned to components of MWUs belonging to a particular inflectional class, and on the other, we identified inflectional classes associated with the same additional information.

B. The rule design strategy

The procedure for automatic construction of a DELAC type dictionary relies on a manually produced set of rules. The rule design strategy resulted from the aforementioned expert analysis of available MWU lemmas. The task of the rule based procedure is to automatically generate the complete MWU lemma. However, the strategy and the procedure are independent, and changes in the strategy, in general, do not affect the procedure itself. This approach enabled us to experiment with various rule strategies, and thus the final strategy used is a result of several iterations.

Our rule based strategy presently consists of 99 rules — 79 for nouns and 20 for adjectives. Among them, 33 rules are for MWUs with 2 components, 34 rules for MWUs with 3 components, 19 rules for MWUs with 4 components, 8 rules for MWUs with 5 components, and 5 rules for MWUs with 6 and 7 components. Examples of two rules are given in Tables II and III.

Conditions defined for each rule are of two types: conditions that specify grammatical categories of MWU components and usually apply to components that inflect, and additional conditions related to semantic and/or syntactic markers of the components. The rule in Table II applies to two-component MWUs, in which the first component is an adjective, the second component is a noun, and the MWU does not inflect in number.

This rule is applied as follows: if the first component satisfies (according to the dictionary of simple words) the specified grammatical conditions, namely, that it is an adjective in the nominative case, and the second component also satisfies (according to the dictionary of simple words) the specified

TABLE II
XML FORM OF A RULE FOR THE CLASS NC_AXN3, SUPER-CLASS NC_AXN

```
<Rule ID='2' CFLX='NC_AXN3' CflxGroup='NC_AXN'>
<RuleGenCond>
    <Word ID='1' POS='A' Flex='true'
        Case='1' Anim='$a' Gen='$g'/>
    <Word ID='2' POS='N' Flex='true'
        Case='1' Anim='=$a' Gen='=$g />
</RuleGenCond>
<RuleSpecCond ID='1' Example='Ajfelova kula'>
    <Word ID='1' Num='s' Cond='$PRE'/>
    <Word ID='2' Num='s'/>
</RuleSpecCond><RuleSpecCond ID='2'
     Example='poljski radovi'>
    <Word ID='1' Case='1' Num='p'/>
    <Word ID='2' Case='1' Num='p'/>
</RuleSpecCond><RuleSpecCond ID='3'
     Example='poljsko cvece'>
    <Word ID='1' Case='1' Num='s'/>
    <Word ID='2' Case='1' Num='s'
        SinSem='+VN,+Coll,+HumColl'/>
</RuleSpecCond>
</Rule>
```

grammatical conditions, namely, that it is a noun in the nominative case, and these two components agree in gender and animacy, then the additional conditions are checked, and at least one of them needs to be satisfied. In this case it means that one of the following additional conditions must be satisfied: the first component starts with uppercase letter (e.g. *Ajfelova kula* 'Eiffel tower'), or both components are already in plural (e.g. *poljski radovi* 'field works'), or the second component is a collective noun (e.g. *poljsko cveće* 'wild flowers').

Another rule that applies to three-component MWU adjectives in the form of a simple word adjective followed by the conjunction *kao*, followed by an animate noun, is given in Table III. An example is the adjective *lukav kao lisica* 'cunning as a fox'. Adjectives of this type have two plural forms: the noun component can be either in the singular *lukavi kao lisica* or in the plural *lukavi kao lisice*. This rule has no additional conditions. Note that in this case the gender of the noun is of no relevance and it need not agree with the gender of the adjective. Namely, feminine case nouns, as the generic name of a zoological species in this case, can be used to describe masculine case nouns.

C. Software implementation

To manipulate the strategy in the form of a XML document our tool LeXimir relies on W3C standard languages Xquery and XSLT supported by .Net. The user interface for automatic production of DELAC lemmas is very straightforward and easy to use. A user can choose a file with a prepared list of MWUs and a file with a strategy, and the results will be presented to him in the form of a table (see Fig. 5) in which the user has only to check the correct solutions upon which a list of DELAC entries is produced.

Figure 5 depicts the resulting table for a list of 8 MWUs.

TABLE III
XML FORM OF A RULE FOR THE CLASS AC_A3XN2, SUPER-CLASS AC_A3XN

```
<Rule ID='153' CFLX='AC_A3XN2' CflxGroup='AC_A3XN'>
<RuleGenCondExample='lukav kao lisica'>
    <Word ID="1" POS='A' Flex='true'
        Case='1' Num='s' Gen='m'/>
    <Word ID="2" POS='MOT' Flex='false'
        Cond='=,kao"/>
    <Word ID="3" POS='N,A' Flex='true'
        Case='1' Num='s' Anim='v'/>
</RuleGenCond>
</Rule>
```

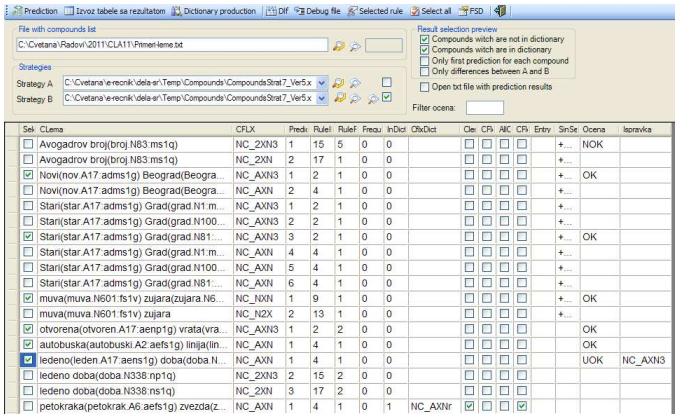


Fig. 5. The Implementation of the Strategy on the prepared list of MWUs

The options offered by the strategy for the first MWU, *Avogadrov broj* 'Avogadro's number', are far from the correct solution, due to the fact that the possessive adjective *Avogadrov* is not included in the Serbian DELAS dictionary of adjectives. As for the second MWU, *Novi Beograd* 'New Belgrade (a municipality of Belgrade)', the first of the two options offered by the strategy is the correct solution. For the third MWU, *Stari Grad* 'Old City (a municipality of Belgrade)' the strategy offers as much as 6 options, among which the third represents the correct solution. Such a large number of options offered is due to the fact that the form *grad* can represent as much as three lemmas: city, degree, and hail. Out of the two options offered by the strategy for the fourth MWU, *muva zujara* 'blow fly', the first one is the correct one. As for the 5th and 6th MWUs, *otvorena vrata* 'open door (a meeting of parents with teachers)' and *autobuska linija* 'bus line' only one solution is offered for each of them, and it is correct in both cases. Three possible solutions are offered for the 7th MWU, *ledeno doba* 'ice age', and one of them, the first, AXN, is partly correct. Namely, the super-class is properly determined, and hence the lemma form, and what remains is to replace the inflection transducer by AXN3, as this MWU does not have a plural. The correction can be made by the user by stating the new, correct name of the transducer in the last column of this partly correct solution. The 8th MWU, *petokraka zvezda* is already in the dictionary which is evidenced by the fact that the column 'ClfxDic', and the following four columns are already filled. The solution offered by the strategy is almost the same as the one existing in the dictionary, except for the fact that the strategy failed to identify that this MWU allows a reversed order of components, which is a highly exceptional feature. The option of the user interface to detect MWUs already in the dictionary is very useful, as it prevents the introduction of duplicates in the dictionary. In addition to that, it may alert the user as to the potential shortcomings of the strategy. When all options offered by the strategy are reviewed, the system will automatically generate lemmas for the DELAC dictionary. In some rare cases all rules will fail and a solution — compound lemma — will not be offered to the user. In that cases a user will have to produce a lemma consulting the excisting e-dicitonary, as illustrated in Figure 4. Thus, we obtain an automated answer to questions 2 and 3 posed at the end of Section II. Question 1 is answered by the user, who prepares the list of input lemmas.

There are various debugging tools and preference selections at user's disposal. In the strategy development phase the user can compare the results obtained by the use of various strategies on the same MWU input list. The user may also filter the results and obtain only those that differ from the results obtained by the previous version of the strategy.

LeXimir has been successfully used for languages other than Serbian and English, namely, for Bulgarian [12]. The new functionality for production of DELAC entries is also expected to perform successfully without any modifications for other languages. The prerequisites are that there exists a Unitex module for that language including: a dictionary of simple words in DELAS format, transducers for the inflection of simple words, the automatically produced dictionary of simple word forms DELAF, and transducers for the inflection of MWUs. As mentioned before, most of these conditions are satisfied for many languages. However, in order to apply this functionality to a new language it would be necessary to develop a new language-dependent strategy, that is, a new XML document. It is also worth mentioning that the system can be easily modified to work with formats of simple words dictionaries other than those supported by Unitex. To that end, only the dictionary look-up module would have to be changed.

D. Procedure Evaluation

In order to evaluate the performance of LeXimir's functionality for automated generation of MWU lemmas, and hence our procedure and our strategy, we have performed experiments on two types of data. The first set of data consisted of nouns and adjectives already available in the existing DELAC dictionaries. The MWU lemmas for dictionary entries were (re)produced by LeXimir and then compared to the (correct) dictionary lemmas. The second set of data consisted of MWUs compiled from several sources, all of them nouns. In both cases the results produced by the system were validated manually.

In line with the possibility of a "partly" correct solution that we have recognized in subsections IV-B and IV-C, the evaluation results were classified as follows:

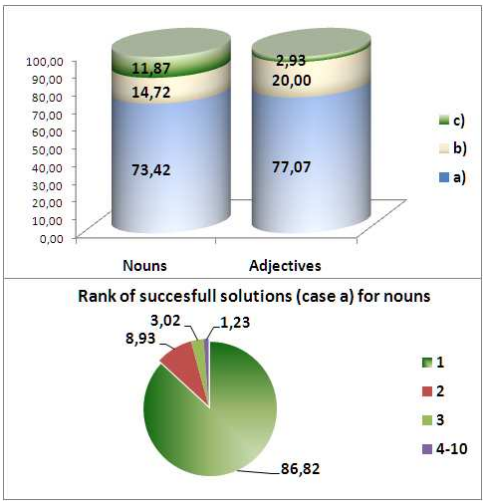


Fig. 6. Results obtained on the first set of test data (in%)

a) If the system produced the correct lemma and assigned the correct inflectional class for a given MWU the overall solution was considered as correct;
b) If the system produced the correct lemma but failed to assign the correct inflectional class, whereas the assigned super-class was correct, the overall solution was considered as partly correct;
c) In all other cases the solution was considered to be incorrect.

As we have already seen, our system can produce more options among which one can be the correct or partly correct solution. In both cases, another point of interest for evaluation was the rank of this (partly) correct option. The most favorable outcome is obviously that this option is the first one on the list. In Fig. 6 we illustrate some of the results for the first set of data (about 2800 existing dictionary entries for nouns and adjectives). The top part of the figure shows the percent of correct solutions for nouns and adjectives produced by the system (case a), the percent of partly correct solutions (case b), and the percent of incorrect solutions (case c). The bottom part of Fig. 6 illustrates the rank of the correct solution in the case of nouns, expressed again in percentages. Namely, for 86,82% percent of MWUs, when the correct option was found it was at the same time the first one offered, whereas for 8,93% MWUs it was the second offered. For less than 5% the correct option was offered at the third or some lower place.

We also performed a more in-depth analysis of the incorrect solutions produced by the experiment with the first set of data. This analysis showed that in the majority of cases (80%) the incorrect solution was due to the fact that one of the MWU components was not in the dictionary of simple words. This happened mainly either because one of more components of a MWU representing a proper name are not words in Serbian, as in *Bab-el-Mandeb*, or because some words are used only in MWUs (like *nagazna* in *nagazna mina* 'landmine'). In both cases there was no justification for including such words in dictionaries of simple words. In a much smaller number of

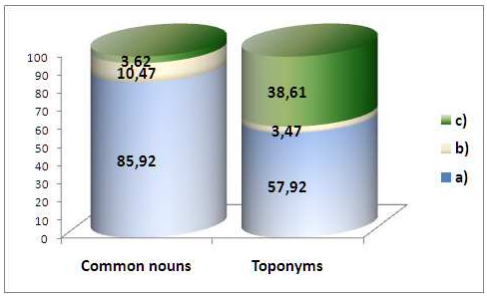


Fig. 7. Results obtained on the second set of test data (in%)

cases (20%) the incorrect solution resulted from the system's failure to cover a specific MWU structure.

With the second set of test data (nouns collected from various sources) we proceeded as follows. First we removed all MWUs that already existed in DELAC which resulted in a list of approximately 1000 MWUs. We separated the list into proper names or toponyms (about 20%) and common nouns (about 80%). The rationale for such an approach was the fact, indicated by the analysis of the first set of data, that system performance tends to decrease considerably in the case of toponyms.

The results illustrated by Fig. 7 confirm the conclusion that toponyms can be viewed as the system's week point. Namely, the system failed to give a correct or partly correct solution for only 3.62% of common nouns, whereas for toponyms this percentage amounts to as much as 38.61%. All of the failures in the case of toponyms resulted from the absence of one or more of its components from dictionaries of simple words in Serbian (e.g. in *Gornji Tavankut*, *Tavankut* is not used independently), which is in line with failure causes in the experiment with the first data set. These lemmas can still be produced within LeXimir following the manual procedure presented in Section III.

Evaluation results are discussed in more detail in [13].

V. Existing and Further Applications

The outlined procedure is now in everyday use for the production of MWU dictionary entries for Serbian. Due to the new functionality implemented in LeXimir the size of the MWU dictionary grew from the initial 2800 lemmas to existing 6450 in a relatively short period. We expect this growth rate to be even greater in the forthcoming period, as many new MWU lists are being prepared.

The benefits obtained by including the MWU dictionary in language processing tasks for Serbian are already clearly visible. Besides the benefits that were to be expected, it has been already shown that the MWU dictionary can also be very useful in text disambiguation [14], and further in the parsing process [15]. We would like to point out another interesting aspect of MWUs which can be exploited in the processing of named entities, as the initial phase in information extraction. Serbian morphological dictionaries and local grammars are successfully being used for recognition of names of persons

and of various functions they might perform within the society. Local grammars for recognition of functions can recognize various syntactic structures but, naturally, not all of them. The use of MWUs can contribute to the increase of the recall without further complicating the local grammars. For example, the local grammar does not recognize the function of the person acting as *specijalni izaslanik UN za pregovore o statusu Kosova Marti Ahtisari* 'UN special envoy for negotiations on the status of Kosovo Martti Ahtisaari' because the addition *o statusu* 'on the status' is not foreseen by the local grammar. When *pregovori o statusu* 'negotiations on the status' are added to the MWU dictionary, the local grammar covers the aforementioned structure as well.

This example leads us to possible applications related to inflection of free noun phrases based on the recognition of their syntactic structure. This idea draws from the assumption that many free noun phrases (used in search queries, for example) may have the same syntactic structure as a MWU, and that the inflectional transducers developed for MWUs could be applied to inflect free noun phrases as well. For example, in the phrase *kućni aparati prošlogodišnje proizvodnje* 'home appliances of last year's production' our procedure would recognize a structure that is inflected according to the AXN4X1 pattern - adjective+noun that do not inflect in number followed by any two words that do not inflect at all.

This approach has already been tested in VeBrana [11]. Namely, as the described procedure for production of DELAC entries was implemented in the core engine of LeXimir it can be used not only in all parts of LeXimir but also in VeBrana, which as we have seen, was in a way built "on top" of LeXimir. This enables expansion of queries submitted to the Google search engine [11]. The main feature of VeBrana is that it enables inflection of simple words, MWUs and free phrases supplied as key-words to Google. The tool relies on Serbian e-dictionaries, inflection transducers for simple words and MWUs, and uses Unitex and Multiflex modules for inflection and dictionary look-up. As for the free phrases that are not in the MWU dictionary, VeBrana relies on its built-in strategy, and always chooses the first of the options offered, which is, as we have seen, the correct one in most cases.

In this context the most interesting issue is the interaction with the user. The interface for query expansion has several levels of complexity in both releases (Windows i.e. standalone and web): the simplest includes only morphological expansion of a query, a more complex one adds synonyms, and the most complex level enables the user to expand his/her query in several ways. For instance, if the initial query is 'marka' and a user chooses to semantically expand his/her query with Serbian wordnet then the system will find, among others, two synsets with appropriate literals: {*marka, zaštitni znak, brend*} 'trade name' and {*marka, poštanska marka*} 'postage stamp'. If MWU synset literals are in the DELAC dictionary (*poštanska marka*), the system directly produces all inflected forms, but if the literal is not yet in the DELAC dictionary (*zaštitni znak*) then the component for production of DELAC forms described in Section IV is invoked to detect its structure, which subsequently generates the inflected forms.

Query expansion in the web environment is implemented in a similar way, with different levels for expansion details. VeBrana accepts the query from the user and submits it to the local web service, which then expands the query and forwards it to the Google search engine. To that end the Google AJAX Search API is used, a Java script library which provides for embedding Google searches into web pages or web applications. The abundance of Google services (Web Search, Local Search, Video Search, Blog Search, News Search and Book Search) are used by this library, consisting of simple web objects aimed at performing "inline" search.

Acknowledgment

This research was supported by the Serbian Ministry of Education and Science under the grant #III 47003.

References

[1] B. Courtois and M. Silberztein, *Dictionnaires électroniques du français*. Paris: Larousse, 1990.

[2] S. Paumier, *Unitex 2.1 User Manual*, http://www-igm.univ-mlv.fr/unitex/UnitexManual2.1.pdf, 2011.

[3] M. Silberztein, "Nooj: A Linguistic Annotation System for Corpus Processing," in *Proceedings of HLT/EMNLP on Interactive Demonstrations*, ser. HLT-Demo '05, 2005, pp. 10–11.

[4] M. Gross, "The use of finite automata in the lexical representation of natural language," in *Electronic dictionaries and automata in computational linguistics*, ser. Lecture Notes in Computer Science, vol. 377. Springer, 1989, pp. 34–50.

[5] C. Krstev, *Processing of Serbian — Automata, Texts and Electronic Dictionaries*. Belgrade: Faculty of Philology, University of Belgrade, 2008.

[6] A. Savary, "Computational Inflection of Multi-Word Units — A Contrastive Study of Lexical Approaches," *Linguistic Issues in Language Technologies*, vol. 1, no. 2, 2008.

[7] C. Krstev and D. Vitas, "Finite State Transducers for Recognition and Generation of Compound Words," in *IS-LTC 2006*, T. Erjavec and J. Žganec Gros, Eds. Ljubljana, Slovenia: Institut "Jožef Stefan", October 2006, pp. 192–197.

[8] A. Savary, "Multiflex: A Multilingual Finite-state Tool for Multi-Word Units," in *CIAA*, 2009, pp. 237–240.

[9] A. Savary, C. Krstev, and D. Vitas, "Inflectional Non-compositionality and Variation of Compounds in French, Polish and Serbian, and Their Automatic Processing," *Bulag — Bulletin de Linguistique Appliquée et Générale*, vol. 32, pp. 73–94, 2007.

[10] A. Savary, J. Rabiega-Wisniewska, and M. Wolinski, "Inflection of Polish Multi-Word Proper Names with Morfeusz and Multiflex," in *Aspects of Natural Language Processing*, ser. Lecture Notes in Computer Science, vol. 5070. Springer, 2009, pp. 111–141.

[11] C. Krstev, R. Stanković, D. Vitas, and I. Obradović, "The Usage of Various Lexical Resources and Tools to Improve the Performance of Web Search Engines," in *6th LREC*, Marrakech, Marocco, 2008.

[12] C. Krstev, R. Stanković, D. Vitas, and S. Koeva, "E-Connecting Balkan Languages," in *Proc. of the Workshop on Multilingual Resources, Technologies and Evaluation for Central and Eastern European Languages — RANLP09*, Borovetz, Bulgaria, 2009, pp. 23–29.

[13] C. Krstev, R. Stanković, I. Obradović, D. Vitas, and M. Utvić, "Automatic Construction of a Morphological Dictionary of Multi-Word Units," in *IceTAL*. Reykavik, Iceland: Springer, August 2010, pp. 226–237.

[14] I. Alegria, O. Ansa, X. Artola, N. Ezeiza, K. Nojenola, and R. Urizar, "Representation and Treatment of Multiword Expressions in Basque," in *Second ACL Workshop on Multiword Expressions: Integrating Processing*, Barcelona, Spain, 2004, pp. 48–55.

[15] E. Wehrli, V. Seretan, and L. Nerima, "Sentence Analysis and Collocation Identification," in *Proc. of the Multiword Expressions: From Theory to Applications — MWE 2010*, Beijing, China, 2010, pp. 28–36.